

#10
KW-8
10-04-02
10/3

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicants:	James Q. Mi, et al.	§	Group Art Unit:	2132
Serial No.:	09/259,620	§		
Filed:	February 26, 1999	§	Examiner:	Douglas J. Meislahn
For:	COMPUTER SYSTEM IDENTIFICATION	§	Atty. Dkt. No.:	ITL.0160US

Board of Patent Appeals & Interferences
Commissioner for Patents
Washington, D.C. 20231

APPEAL BRIEF

Dear Sir:

Applicant hereby appeals from the Final Rejection dated July 22, 2002, finally
rejecting claims 1-26.

I. REAL PARTY IN INTEREST

The real party in interest is Intel Corporation, the assignee of the present
application by virtue of the assignment recorded at Reel/Frame 9797/0271.

II. RELATED APPEALS AND INTERFERENCES

There are no related appeals or interferences.

RECEIVED
SEP 23 PM 1:58
BOARD OF PATENT APPEALS
AND INTERFERENCES

Date of Deposit: September 17, 2002
I hereby certify under 37 CFR 1.8(a) that this
correspondence is being deposited with the United States
Postal Service as **first class mail** with sufficient postage
on the date indicated above and is addressed to the Board
of Patent Appeals & Interferences, Commissioner for
Patents, Washington, DC 20231.
Debra Cutrona
Debra Cutrona

III. STATUS OF THE CLAIMS

The application was originally filed with claims 1-20, and claims 21-26 were added during prosecution of the application. Claims 1-26 have been finally rejected under 35 U.S.C. § 103(a), and these rejections are the subject of this appeal. Claim 23 was rejected in the Final Office Action under 35 U.S.C. § 112 for referring to a first computer system of claim 7 rather than the apparatus of claim 7. The Examiner indicated in the Advisory Action that the amendment had been entered. The Examiner did not indicate in the Advisory Action whether or not the § 112 rejection had been withdrawn. Because this amendment conformed to the amendment that was suggested by the Examiner in the Final Office Action, it is assumed for purposes of this appeal that the § 112 rejection of claim 23 has been withdrawn.

IV. STATUS OF AMENDMENTS

There are no unentered amendments.

V. SUMMARY OF THE INVENTION

Referring to Fig. 1, an embodiment 10 of a computer system in accordance with the invention includes an encryption unit 31 that may receive identification requests from web sites 36 (web sites 36a, 36b and 36c, as examples) for an identity of the computer system 10. In response to these requests, the encryption unit 31 may furnish different hash values 32 (hash values 32a, 32b and 32c, as examples) to the different web sites 36. In some embodiments, each hash value 32 is different, and as a result, each web site 36 may identify the computer system 10 by a different hash value 32, although each of the

hash values 32 is generated by a single processor number 30, as described below.

Because each web site 36 associates the computer system 10 with a different hash value 32, information about a user of the computer system 10 may not be correlated between databases that are maintained by different web sites 36. For example, a particular web site 36 may identify the computer system 10 via the hash value “1bdf23” and another web site 36 may identify the computer system 10 via the hash value “53gh44.”

Furthermore, as described below, the manner in which the encryption unit 31 generates the hash values 32 makes it very difficult for a rogue web site 36 from obtaining the hash values 32 that identify the computer system 10 to other web sites 36. Therefore, due to the technique used by the encryption unit 31, it may be very different to correlate information about the user from databases that are maintained by different web sites 36. In this context, the term “web site” generally refers to an arrangement where a computer system (a server, for example) executes software to provide a service to other computer systems, such as the computer system 10. Specification, pp. 3-4.

In the context of this application, the phrase “computer system” may generally refer to a processor-based system and may include (but is not limited to) a graphics system, a desktop computer, a mobile computer (a laptop computer, for example), or a set-top box as just a few examples. The term “processor” may refer to, as examples, at least one central processing unit (CPU), microcontroller, X86 microprocessor, Advanced RISC Machine (ARM) microprocessor or Pentium-based microprocessor. The examples listed above are not intended to be limiting, but rather, other types of computer systems

and other types of processors may be included in some embodiments of the invention.

Specification, p. 4.

To obtain a hash value 32 that identifies the computer system 10, a particular web site 36 may transmit a privacy key 34 (privacy keys 34a, 34b and 34c, as examples) to the computer system 10. In response, the encryption unit 31 may encrypt an embedded identifier, such as a processor number 30, with the privacy key 34 to produce the hash value 32 that the computer system 10 furnishes to the requesting web site 36. In this manner, if each web site 36 transmits a different privacy key 34 to the computer system 10, then each web site 36 receives a different hash value 32, each of which indicates the computer system 10 to the particular web site 36. As described further below, the encryption unit 31 may include a processor 200 (see Fig. 3) to aid in the encryption of the privacy key 34 with the processor number 30. Specification, pp. 4-5.

The privacy key 34 may or may not be a private key, depending on the particular embodiment. For example, in some embodiments, the privacy key 34 may be derived from an address or universal resource locator (URL) for the web site 36. Therefore, as an example, the privacy key 34 may indicate a string, such as "www.example.com." As described below, for the embodiments where the privacy key 34 is derived from the URL, the computer system 10 may perform a validity check to determine if the privacy key 34 that is furnished by a particular web site 36 is based on the URL of the web site 36.

Specification, p. 5.

In some embodiments, the encryption unit 31 may use a hash function called $F(PN, PRIVACYKEY)$ to perform the encryption. The $F(PN, PRIVACYKEY)$ function

may have properties that make it more difficult to track user information (about the computer system 10) that is stored on different web sites 36. For the $F(\text{PN}, \text{PRIVACYKEY})$ hash function, the notation “PN” represents the processor number 30, and the notation “PRIVACYKEY” represents the privacy key 34. Specification, p. 5.

One of the properties of the $F(\text{PN}, \text{PRIVACYKEY})$ hash function may be that the $F(\text{PN}, \text{PRIVACYKEY})$ function is a one way hash function, a notation that implies given the hash value 32 and the privacy key 34, it may be very difficult, if not impossible, to work backwards to determine the processor number 30. As a result, it may be very difficult for a particular web site 36 to use the hash value 32 that is obtained by that web site 36 to derive the processor number 30. Specification, p. 5.

In some embodiments, another property of the $F(\text{PN}, \text{PRIVACYKEY})$ function may be that the $F(\text{PN}, \text{PRIVACYKEY})$ function is collision free, a term that means that it is highly unlikely for the $F(\text{PN}, \text{PRIVACYKEY})$ hash function to return the same hash value for different privacy keys 34. Thus, it may be highly unlikely for a particular website 36 to use the $F(\text{PN}, \text{PRIVACYKEY})$ function (with its associated privacy key 34) to obtain the same hash value 32 for two different processor numbers 30. Thus, this feature ensures that it is highly likely for a particular web site 36 to identify each computer system with a different, unique processor number 30. Specification, pp. 5-6.

Yet another property of the $F(\text{PN}, \text{PRIVACYKEY})$ function (in some embodiments) may be that the $F(\text{PN}, \text{PRIVACYKEY})$ function is non-commutative, as described below:

$$F(F(\text{PN}, \text{PRIVACYKEY}), \text{PRIVACYKEY}') \neq$$

$F(F(PN, \text{PRIVACYKEY}'), \text{PRIVACYKEY}))$,

where “PRIVACYKEY’” represents a privacy key 34 that is different from the privacy key 34 that is represented by “PRIVACYKEY.” As a result of the non-commutative property, it may be very difficult to correlate the information that is associated with the computer system 10 (and user) on different databases (on different web sites 36) when different privacy keys 34 are used. Specification, p. 6.

Many different hash functions may be used, in various embodiments, that satisfy one, more than one, or all of the properties described below. For example, in some embodiments, a secure hash algorithm (SHA), an algorithm that satisfies all of the properties described above, may be used. Specification, p. 6.

In some embodiments, the computer system 10 may notify the user of the system 10 when a particular web site 36 is requesting system identification. For example, this notification may be in the form of a prompt in a window that is formed on a display 14 (see Fig. 3) of the computer system 10. In this manner, the user may either permit the web site 36 to obtain the identification (provided by the hash value 32) or reject the request. In some embodiments, the user may have an option to turn off the prompt. Specification, p. 6.

Besides prompting the user about the identification request, the computer system 10 may take measures to prevent a rogue web site 36 from submitting an incorrect privacy key 34 for purposes of obtaining a hash value 32 that is associated with another web site 36. For example, in some embodiments, the request for identification may involve a two-part identification procedure. First, the web site 36 sets the privacy key 34

by executing (if authorized, as described below) an instruction (called SETKEY(PRIVACYKEY)) of the processor 200 (see Fig. 2). Referring to Fig. 2, as described below, the SETKEY(PRIVACYKEY) function may be associated with ring zero (i.e., the highest level) of an operating system 28. As a result, the computer system 10 may not permit execution of this processor instruction until the computer system 10 validates the provided privacy key 34 by executing a software program called a driver 19. After the privacy key 34 is validated by execution of the driver 19, the web site 36 may then be authorized to execute a processor instruction called HWID() (i.e., the HWID() instruction may not have an input parameter) that is associated with ring three (i.e., a lower privilege level) of the operating system 28 to obtain the hash value 32. Specification, pp. 6-7.

More particularly, in some embodiments, the above-described identification procedure may involve interaction between the operating system 28, an Internet browser 27 (Internet Explorer ® or Netscape Navigator ®, as examples) and the driver 19. For example, because the SETKEY(PRIVACYKEY) instruction is associated with ring zero, the web site 36 may not by itself cause execution of the instruction to obtain the hash value 32, as the web site 36 may only have access to ring three (a lower privilege level) and higher rings (i.e., even lower privilege levels) of the operating system 28. However, the driver 19 may have ring zero privileges and thus, may form a bridge between the web site 36 and the ring zero privileges of the operating system 28. In this manner, when the web site 36 attempts to execute SETKEY(PRIVACYKEY) instruction, the driver 19 may be called by the operating system 28 to cause the processor 200 to validate the privacy

key 34 before providing the hash value 32. In the execution of the driver 19, the processor 200 may use results obtained from the execution of the browser 27 to validate the privacy key 34, as described below. Specification, p. 7.

Referring to Fig. 4, when executed by the processor 200, the driver 19 may cause the processor 200 to perform the following functions. In particular, the driver 19 may cause the processor 200 to determine (diamond 50) if the user has enabled an option to prompt the user when an identification request is received. If so, the processor 200 prompts (block 52) the user (via the display 14 (see Fig. 2), for example) that a web site 36 has submitted an identification request and waits for the user to indicate (via a keyboard 24 or move 26 (see Fig. 2), as examples) whether the user desires to reject the request. If so, the processor 200 rejects the request by notifying (block 56) the web site 36. Specification, pp. 7-8.

However, if the user did not reject the request, then the processor 200 may determine (diamond 58) whether the browser 27 is currently being executed. If so, the program 19 causes the processor 200 to communicate (block 60) the privacy key 34 to the browser 27 so that when the processor 200 executes the browser 27 (on another thread, for example), the processor 200 may compare the URL of the web site 32 to the privacy key 34. Subsequently, the processor 200, communicates the results of the comparison for use by the driver 19. In this manner, when the processor 200 subsequently executes the driver 19, the processor 200 determines (diamond 62) whether the privacy key 34 matches the URL of the web site 36. If not, the processor 200 rejects the request and notifies (block 56) the web site 36 about the rejection of the identification

request. Otherwise, the processor 200 executes (block 64) the SETKEY(PRIVACYKEY) instruction to set the privacy key to be used for the encryption of the processor number 30. In this manner, the web site 36 that submitted the privacy key 34 may cause the processor 200 to execute the HWID() instruction to cause the processor 200 to produce an indication of the hash value 32. However, if the privacy key 34 has not been set, then the processor 200 returns an indication of an error rather than the indication of the hash value 32. Specification, p. 8.

Referring back to Fig. 3, in some embodiments, the computer system 10 may include a bridge, or memory hub 16. The processor 200 and the memory hub 16 may be coupled to a host bus 23. The memory hub 16 may provide interfaces to couple the host bus 23, a memory bus 29 and an Accelerated Graphics Port (AGP) bus 11 together. The AGP is described in detail in the Accelerated Graphics Port Interface Specification, Revision 1.0, published on July 31, 1996, by Intel Corporation of Santa Clara, California. The system memory 18 may be coupled to the memory bus 29, and store the driver 19, the browser 27 and portions of the operating system 28 (not shown in Fig. 3). A graphics accelerator 13 (that controls the display 14) may be coupled to the AGP bus 11. A hub communication link 15 may couple the memory hub 16 to another bridge circuit, or input/output (I/O) hub 20. Specification, pp. 8-9.

In some embodiments, the I/O hub 20 includes interfaces to an I/O expansion bus 25 and a Peripheral Component Interconnect (PCI) bus 21. The PCI Specification is available from The PCI Special Interest Group, Portland, Oregon 97214. A network interface 12 (a modem or a local area network (LAN) card, as examples) may be coupled

to the PCI bus 21 and provide a communication path for the computer system 10 to communicate with the web sites 36. In this manner, the processor 200 may interact with the network interface 12 to communicate with the web sites 32. The I/O hub 20 may also include interfaces to a hard disk drive 37 and a CD-ROM drive 33, as examples. An I/O controller 17 may be coupled to the I/O expansion bus 25 and receive input data from the keyboard 24 and the mouse 26, as examples. The I/O controller 17 may also control operations of a floppy disk drive 22. Copies of the driver 19 may be stored on, as examples, the hard disk drive 32, a diskette or a CD-ROM, as just a few examples.

Specification, p. 9.

Referring to Fig. 5, as an example, the processor 200 may include a bus interface unit (BIU) 208 that is coupled to address, control and data lines of the host bus 23 to, among other operations, retrieve instructions and data from the system memory 18. For the instructions, the processor 19 may include an instruction unit 203 that is coupled to the bus unit 208 to decode the instructions. In this manner, the instruction unit 203 may have buffers and a cache to store the instructions. A control unit 208 (of the processor 200) may receive signals from the instruction unit 203 that indicate the decoded instructions. For example, the signals may indicate the instruction to perform the SETKEY(PRIVACYKEY) function or the instruction to perform the HWID() function.

Specification, p. 9.

In response to the instruction that is indicated by the instruction unit 203, in some embodiments, the control unit 208 may retrieve corresponding elementary instructions, called microcode, from a microcode read only memory (ROM) 210 of the processor 200

and execute the microcode. For example, microcode 211 to cause the processor 200 to perform the SETKEY(PRIVACYKEY) and HWID() instructions may be stored in a microcode read only memory (ROM) 210. In performing the execution of an instruction, the control unit 208 may control an arithmetic logic unit (ALU) 212, registers 214 and an addressing unit 206. Specification, pp. 9-10.

In other embodiments, the circuitry to perform the SETKEY(PRIVACYKEY) and HWID() instructions may be hardwired instead of being implemented in microcode. The processor number 30 may be replaced by another identifier that identifies the computer system 10. A privacy key other than a string that indicates an URL may be used. Applications other than applications being executed by web sites may request identification of the computer system 10. For example, other computer systems that are connected through a local area network (LAN) may request identification from the computer system 10. Specification, p. 10.

VI. ISSUES

- A. **Can claims 1-5, 21 and 22 be rendered obvious when the Examiner has failed to establish a *prima facie* case of obviousness for claim 1?**
- B. **Can claims 21 and 22 be rendered obvious when the Examiner has failed to establish a *prima facie* case of obviousness for claim 21?**
- C. **Can claim 22 be rendered obvious when the Examiner has failed to establish a *prima facie* case of obviousness for claim 22?**
- D. **Can claims 6-9, 23 and 24 be rendered obvious when the Examiner has failed to establish a *prima facie* case of obviousness for claim 6?**
- E. **Can claims 23 and 24 be rendered obvious when the Examiner has failed to establish a *prima facie* case of obviousness for claim 23?**

- F. Can claim 24 be rendered obvious when the Examiner has failed to establish a *prima facie* case of obviousness for claim 24?**
- G. Can claims 10-14, 25 and 26 be rendered obvious when the Examiner has failed to establish a *prima facie* case of obviousness for claim 10?**
- H. Can claims 25 and 26 be rendered obvious when the Examiner has failed to establish a *prima facie* case of obviousness for claim 25?**
- I. Can claim 26 be rendered obvious when the Examiner has failed to establish a *prima facie* case of obviousness for claim 26?**
- J. Can claims 15-20 be rendered obvious when the Examiner has failed to establish a *prima facie* case of obviousness for claim 15?**

VII. GROUPING OF THE CLAIMS

Claims 1-5 can be grouped together; claims 6-9 can be grouped together; claims 10-14 can be grouped together; claims 15-20 can be grouped together; and claims 21-26 are each separately patentable for the reasons set forth below.

VIII. ARGUMENT

All claims should be allowed over the cited references for the reasons set forth below.

- A. Can claims 1-5, 21 and 22 be rendered obvious when the Examiner has failed to establish a *prima facie* case of obviousness for claim 1?**

The method of independent claim 1 includes receiving a request from a first computer system for identification of a second computer system, retrieving an identifier that identifies the second computer system and encrypting the identifier with a key that is associated with the first computer system to produce a hash value. The hash value is provided to the first computer system in response to the request.

The Examiner rejects independent claim 1 under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 5,825,884 (herein called "Zdepski") in view of the reference book Schneier, Bruce, *Applied Cryptography*, John Wiley and Sons, Inc. (1996) (herein called "Schneier"). Zdepski generally discloses a server system for transferring subscriber information requests to information service providers. Schneier generally discloses cryptographic techniques.

More specifically, Zdepski teaches one-way communication between a subscriber platform 145 and a transactional server system 270. In this manner, the subscriber platform 145 transmits an information request 275 that includes an encrypted subscriber identification and a message request block. Zdepski, 4:16-22. Zdepski neither teaches nor suggests that the subscriber platform 145 furnishes the subscriber identification in response to a request from another computer system (such as the transactional server system 270, for example) for identification of the platform 145. The only transmission from the transactional server system 270 to the subscriber platform 145 in regards to the subscriber identification communication is an acknowledgment that the transmission of the subscriber identification and the message request block has been received. Zdepski, 4:44-46. Such an acknowledgment does not constitute a request for the subscriber identification.

Referring back to the claim language, the method of claim 1 sets forth receiving a request from a first computer system for identification of a second computer system. Zdepski neither teaches nor suggests receiving such a request, as neither the subscriber platform 145 nor any other entity receives such a request.

The Examiner contends that it would have been obvious to modify Zdepski in view of Schneier. However, the Examiner fails to provide support for such a motivation or suggestion for this modification. The only attempt to provide support for such a motivation occurs in the Advisory Action, in which the Examiner states that the support appears in lines 64-67 of column 4 of Zdepski. Referring to this cited passage, this language discusses encrypting a subscriber identification with a database server public key. However, this language does not serve as a suggestion or motivation to modify Zdepski so that the transactional server 270 or some other computer system requests identification of the subscriber platform 145. To the contrary, there is no motivation or suggestion to modify Zdepski so that a computer system (such as the transactional server system 270) requests the identification of the subscriber platform 145 because this identify is already presumed, as the communication of the subscriber identification does not occur in response to a request of this identification from the transactional server system 270.

"Obviousness cannot be predicated on what is unknown." *In re Spormann*, 363 F2d 444, 448, 150 USPQ 449, 452 (CCPA 1966). Thus, for purposes of establishing a *prima facie* case of obviousness, the Examiner must provide support for the alleged suggestion or motivation for the modification of Zdepski. This means that the Examiner must point out specific language of Zdepski, Schneier or another prior art reference to support the alleged suggestion or motivation to modify Zdepski in view of Schneier. *Ex parte Gambogi*, 62 USPQ2d 1209, 1212 (Bd. Pat. App. & Int. 2001); *In re Rijckaert*, 28 USPQ2d 1955, 1957 (Fed. Cir. 1993); M.P.E.P. § 2143. As discussed above, the

language cited by the Examiner in the Advisory Action does not constitute such a suggestion or motivation.

Thus, for at least the reason that the Examiner has not provided support for the alleged suggestion or motivation to modify Zdepski in view of Schneier, a *prima facie* case of obviousness for independent claim 1 has not been established.

A *prima facie* case of obviousness for claim 1 has not been established for at least the additional reason that neither Zdepski nor Schneier teaches or suggests retrieving an identifier that identifies a second computer system. The Examiner contends that the encryption of a "subscriber identification" (in lines 63-67 of column 4) teaches retrieving an identifier that identifies a second computer system. However, Zdepski does not specify what is meant by "subscriber identification." In this manner, "subscriber identification" is ambiguous and could refer to an account number of a person subscribing to a service and not refer to a particular computer system, for example. Thus, "subscriber identification" does not specifically refer to the identification of a particular computer system, as set forth in claim 1.

The Examiner contends, "the subscriber platform and identification are both associated with the subscriber and hence associated with each other," and further adds, "as such, the subscriber identification identifies the subscriber platform." Final Office Action, p. 3. However, this conclusion is unsupported by Zdepski, as Zdepski neither teaches nor suggests that the subscriber identification identifies a subscriber platform or that the subscriber identification is somehow correlated to the subscriber platform. As noted above, "obviousness cannot be predicated on what is unknown." *In re Spormann*,

150 USPQ at 452. Therefore, for at least the additional reason that none of the cited references teach or suggest retrieving an identifier that identifies a second computer system, a *prima facie* case of obviousness has not been established for independent claim 1.

Dependent claims 2-5, 21 and 22 are patentable for at least the reason that these claims depend from an allowable claim.

Thus, the rejections of claims 1-5, 21 and 22 are improper and should be reversed.

B. Can claims 21 and 22 be rendered obvious when the Examiner has failed to establish a *prima facie* case of obviousness for claim 21?

The method of claim 21 sets forth that the processor number of claim 2 identifies a microprocessor of the second computer system.

The Examiner rejects claim 21 under 35 U.S.C. § 103(a) as being unpatentable over Zdepski and Schneier in view of U.S. Patent No. 5,978,482 (herein called "Dwork"). Dwork generally teaches a method and system for the protection of digital information.

In the § 103 rejection of claim 21, the Examiner takes official notice that the use of microprocessors "is old and well-known" and relies on Dwork for the missing claim limitations. However, even with the recognition that the use of microprocessors is old and well known, Dwork does not supply the missing claim limitations. In fact, the Examiner refers to no language of Dwork that discloses a processor number that identifies a microprocessor. Neither Zdepski nor Schneier teaches or suggests a processor number that identifies a microprocessor.

Therefore, for at least the additional independent reason that none of the cited references teach or suggest a processor number identifying a microprocessor, a *prima facie* case of obviousness has not been established for dependent claim 21. Claim 22 is patentable for at least the reason that this claim depends from claim 21, an allowable claim.

Thus, for at least these additional reasons, the rejections of claims 21 and 22 are improper and should be reversed.

C. Can claim 22 be rendered obvious when the Examiner has failed to establish a *prima facie* case of obviousness for claim 22?

Claim 22 sets forth that the processor number of claim 21 uniquely identifies the microprocessor.

The Examiner rejects claim 22 under 35 U.S.C. § 103(a) as being unpatentable over Zdepski and Schneier in view of Dwork. As set forth above in the discussion of Issue B, the Examiner relies on Dwork to teach a processor number that identifies a microprocessor. The Examiner, in the rejection of claim 22, also relies on Dwork to teach that the processor number uniquely identifies a microprocessor. However, in light of Dwork's failure to teach a processor number that identifies a microprocessor, Dwork further fails to teach a processor number that uniquely identifies a microprocessor. Neither Schneier nor Zdepski teaches or suggests these missing claim limitations.

The Examiner fails to specifically point out where such limitations are shown in Dwork. The recognition by the Examiner that microprocessors are old and well known does not supply the missing claim limitations. Thus, for at least this additional

independent reason, a *prima facie* case of obviousness has not been established for claim 22.

Therefore, the § 103(a) rejection of claim 22 is improper and should be reversed.

D. Can claims 6-9, 23 and 24 be rendered obvious when the Examiner has failed to establish a *prima facie* case of obviousness for claim 6?

The apparatus of independent claim 6 includes an interface and a processor. The interface is adapted to receive a request from a computer system for identification of the apparatus and furnish a hash value that identifies the apparatus to the computer system. The processor is coupled to the interface and is adapted to encrypt an identifier that identifies the apparatus with a key that is associated with the computer system to produce the hash value.

The Examiner rejects independent claim 6 under 35 U.S.C. § 103(a) as being unpatentable over Zdepski in view of Schneier. However, Zdepski does not teach an interface that is adapted to receive a request from a computer system for identification of an apparatus and furnish a hash value that identifies the apparatus to the computer system. As set forth above in the discussion of Issue A, Zdepski effectively discloses one-way communication of the subscriber identification and neither teaches or suggests an interface (of an apparatus) to receive a request from a computer system for identification of the apparatus.

Thus, the Examiner relies on Schneier to supply the missing claim limitations. However, the Examiner fails to establish a *prima facie* case of obviousness because the Examiner does not provide any support for a suggestion or motivation to modify Zdepski

so that the subscriber platform 145 receives requests for identification. In this manner, the Examiner provides no support for the alleged suggestion or motivation to modify Zdepski so that the transactional server system 270 receives a request for its identification. Therefore, for at least this reason, the Examiner fails to establish a *prima facie* case of obviousness for independent claim 6.

The Examiner fails to establish a *prima facie* case of obviousness for independent claim 6 for at least the additional reason that neither Zdepski nor Schneier teaches or suggests an interface that is adapted to furnish a hash value that identifies an apparatus. In this manner, the Examiner refers to Zdepski's teaching of encrypting a subscriber identification as disclosing furnishing a hash value that identifies an apparatus. However, "subscriber identification" is ambiguous and could refer alternatively to an account number of a person that subscribes to the service, for example. Zdepski fails to disclose any correlation between "subscriber identification" and the identity of an apparatus. Thus, Zdepski does not teach furnishing a hash value that identifies an apparatus.

Therefore, for at least this additional reason, the Examiner fails to establish a *prima facie* case of obviousness for independent claim 6. Claims 7-9, 23 and 24 are patentable for at least the reason that these claims depend from an allowable claim.

Thus, the § 103(a) rejections of claims 6-9, 23 and 24 are improper and should be reversed.

E. Can claims 23 and 24 be rendered obvious when the Examiner has failed to establish a *prima facie* case of obviousness for claim 23?

The computer system of claim 23 sets forth that the processor number of claim 7 identifies a microprocessor of the apparatus.

The Examiner rejects claim 23 under 35 U.S.C. § 103(a) as being unpatentable over Zdepski and Schneier in view of Dwork. However, none of these references teaches or suggests a processor number that identifies a microprocessor. In this manner, the recognition by the Examiner that microprocessors are old and well known does not teach or even suggest a processor number that identifies a microprocessor. Claim 24 is patentable for at least the reason that this claim depends from an allowable claim, claim 23.

Thus, for at least these reasons, the § 103(a) rejections of claims 23 and 24 are improper and should be reversed.

F. Can claim 24 be rendered obvious when the Examiner has failed to establish a *prima facie* case of obviousness for claim 24?

The computer system of claim 24 sets forth that the processor number of claim 23 uniquely identifies a microprocessor.

The Examiner rejects claim 24 under 35 U.S.C. § 103(a) as being unpatentable over Zdepski and Schneier in view of Dwork. As discussed above in connection with Issue E, the Examiner relies on Dwork to teach a processor number that identifies a microprocessor. However, Dwork neither teaches nor suggests such a processor number. Thus, without teaching a processor number that identifies a microprocessor, Dwork

cannot teach or suggest a processor number that uniquely identifies a microprocessor. The recognition by the Examiner that microprocessors are old and well known in the art neither teaches nor suggests a processor number that uniquely identifies a microprocessor. Thus, for at least this additional, independent reason, the Examiner fails to establish a *prima facie* case of obviousness for claim 24.

Therefore, for at least this additional reason, the § 103(a) rejection of claim 24 is improper and should be reversed.

G. Can claims 10-14, 25 and 26 be rendered obvious when the Examiner has failed to establish a *prima facie* case of obviousness for claim 10?

The article of independent claim 10 includes a storage medium that is readable by a first processor-based system. The storage medium stores instructions to cause a processor to receive a key from another processor-based system for identifying the first system. The instructions cause the processor to determine whether the key is valid and based on the identification, selectively authorize encryption of an identifier that identifies the first system with the key to produce a hash value.

The Examiner rejects independent claim 10 under 35 U.S.C. § 103(a) in view of Zdepski and Schneier. Claim 10 sets forth that the instructions cause the processor to receive a key from another processor-based system for identifying the first system. However, Zdepski neither teaches nor suggests instructions to cause a processor to receive a key for identifying a first system and based on the identification, selectively authorize an encryption of an identifier that identifies a first processor-based system.

The Examiner relies on Schneier to teach instructions to cause a processor to receive a key from another processor-based system for identifying the system. However, the Examiner provides no support for a suggestion or motivation to modify Zdepski so that the subscriber platform 145 receives such a key. Thus, for at least this reason, the Examiner fails to establish a *prima facie* case of obviousness for independent claim 10.

The Examiner fails to establish a *prima facie* case of obviousness for claim 10 for at least the additional reason that neither Zdepski nor Schneier teaches or suggests an identifier that identifies a first processor-based system. The Examiner relies on the language in lines 63-67 of column 4 of Zdepski to teach such an identifier. However, this language describes a subscriber identification. As discussed above, "subscriber identification" is ambiguous and does not teach or suggest an identifier that identifies a processor-based system. Thus, for at least this additional reason, a *prima facie* case of obviousness has not been established for claim 10.

Claims 11-14, 25 and 26 are patentable for at least the reason that these claims depend from an allowable claim.

Thus, for at least these reasons, the § 103 rejections of claims 10-14, 25 and 26 are improper and should be reversed.

H. Can claims 25 and 26 be rendered obvious when the Examiner has failed to establish a *prima facie* case of obviousness for claim 25?

Claim 25 sets forth that the processor number of claim 14 identifies a microprocessor of the first system.

The Examiner rejects claim 25 under 35 U.S.C. § 103(a) in view of Zdepski and Schneier in view of Dwork. However, none of these references teaches or suggests a processor number that identifies a microprocessor. In this manner, the recognition of the Examiner that microprocessors are old and well known does not teach or even suggest a processor number that identifies a microprocessor. Claim 26 is patentable for at least the reason that this claim depends from an allowable claim, claim 25.

Thus, for at least these additional, independent reasons, the § 103(a) rejections of claims 25 and 26 are improper and should be reversed.

I. Can claim 26 be rendered obvious when the Examiner has failed to establish a *prima facie* case of obviousness for claim 26?

Claim 26 sets forth that the processor number of claim 25 uniquely identifies a microprocessor.

The Examiner rejects claim 26 under 35 U.S.C. § 103(a) in view of Zdepski and Schneier in view of Dwork. The Examiner relies on Dwork to teach a processor number that identifies a microprocessor. However, Dwork neither teaches nor suggests such a processor number. Thus, without teaching a processor number that identifies a microprocessor, Dwork cannot teach or suggest a processor number that uniquely identifies a microprocessor. Thus, for at least this additional, independent reason, the Examiner fails to establish a *prima facie* case of obviousness for claim 26.

Therefore, the § 103(a) rejection of claim 26 is improper and should be reversed.

J. Can claims 15-20 be rendered obvious when the Examiner has failed to establish a *prima facie* case of obviousness for claim 15?

The microprocessor of independent claim 15 includes an instruction unit, an execution unit and a bus interface unit. The instruction unit is adapted to indicate when the instruction unit receives an instruction that requests an identifier that identifies the microprocessor. The execution unit is coupled to the instruction unit and is adapted to, in response to the indication from the instruction unit, encrypt a key with the identifier to produce a hash value. The bus interface is coupled to the execution unit and is adapted to furnish an indication of the hash value to external pins of the microprocessor.

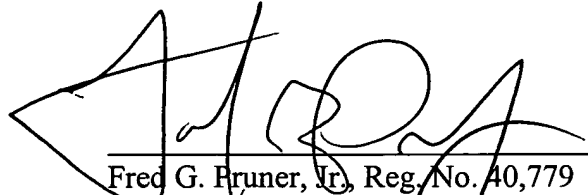
The Examiner rejects independent claim 15 under 35 U.S.C. § 103(a) in view of Zdepski and Schneier. However, the Examiner fails to establish a *prima facie* case of obviousness for claim 15. In this manner, the Examiner fails to show a teaching in either Zdepski or Schneier of an instruction unit that is adapted to indicate when the instruction unit receives an instruction that requests an identifier that identifies a microprocessor. Zdepski only mentions a "subscriber identification," a term that neither teaches nor suggests identification of a microprocessor. Furthermore, Schneier neither teaches nor suggests the missing claim limitations. Thus, because the Examiner fails to show all the claim limitations in Schneier or Zdepski, alone or in combination, a *prima facie* case of obviousness has not been established for independent claim 15. Claims 16-20 are patentable for at least the reason that these claims depend from an allowable claim.

Thus, the § 103(a) rejections of claims 15-20 are improper and should be reversed.

IX. CONCLUSION

Applicant requests that each of the final rejections be reversed and that the claims subject to this appeal be allowed to issue.

Respectfully submitted,



Fred G. Pruner, Jr., Reg. No. 40,779
TROP, PRUNER & HU, P.C.
8554 Katy Freeway, Suite 100
Houston, TX 77024-1805
713/468-8880 [Phone]
713/468-8883 [Facsimile]

Date: September 17, 2002



21906

PATENT TRADEMARK OFFICE

APPENDIX OF CLAIMS

The claims on appeal are:

1. A method comprising:

receiving a request from a first computer system for identification of a second computer system;

retrieving an identifier that identifies the second computer system;

encrypting the identifier with a key associated with the first computer system to produce a hash value; and

providing the hash value to the first computer system in response to the request.
2. The method of claim 1, wherein the act of retrieving the identifier comprises:

retrieving a processor number that identifies a processor of the second computer system.
3. The method of claim 2, further comprising:

executing a processor instruction; and

retrieving the number in response to the execution of the instruction.
4. The method of claim 1, further comprising:

receiving the key from the first computer system.
5. The method of claim 1, wherein the key indicates an address of a web site.
6. An apparatus comprising:

an interface adapted to:

receive a request from a computer system for identification of the apparatus, and

furnish a hash value that identifies the apparatus to the computer system; and

a processor coupled to the interface and adapted to:

encrypt an identifier that identifies the apparatus with a key associated with the computer system to produce the hash value.

7. The apparatus of claim 6, wherein the identifier comprises a processor number that identifies the processor.

8. The apparatus of claim 6, wherein the processor comprises:
a memory adapted to store microcode for performing the encryption; and
a control unit coupled to the memory and adapted to execute the microcode to perform the encryption.

9. The apparatus of claim 6, wherein the processor is further adapted to:
interact with the interface to receive the key from the computer system.

10. An article comprising a storage medium readable by a first processor-based system, the storage medium storing instructions to cause a processor to:
receive a key from another processor-based system for identifying the first system,
determine whether the key is valid,
based on the identification, selectively authorize encryption of an identifier that identifies the first system with the key to produce a hash value.

11. The article of claim 10, the storage medium storing instructions to cause the processor to:
use an address of said another system to determine whether the key is valid.

12. The article of claim 11, wherein the key indicates an URL address.

13. The article of claim 10, the storage medium storing instructions to cause the processor to:

execute an instruction to cause the processor to subsequently use the key to produce the hash value.

14. The article of claim 10, wherein the identifier comprises a processor number.

15. A microprocessor comprising:

an instruction unit adapted to indicate when the instruction unit receives an instruction that requests an identifier that identifies the microprocessor;

an execution unit coupled to the instruction unit and adapted to, in response to the indication from the instruction unit, encrypt a key with the identifier to produce a hash value; and

a bus interface unit coupled to the execution unit and adapted to furnish an indication of the hash value to external pins of the microprocessor.

16. The microprocessor of claim 15, wherein the execution unit comprises:

a control unit; and

a memory coupled to the control unit and storing microcode to cause the control unit to use the key and the identifier to produce the hash value.

17. The microprocessor of claim 15, wherein the identifier comprises a processor number.

18. The microprocessor of claim 15, wherein the execution unit is adapted to use a one way hash function to produce the hash value.

19. The microprocessor claim 15, wherein the execution unit is adapted to use a non-commutative hash function to produce the hash value.

20. The microprocessor of claim 15, wherein the execution unit is adapted to use a collision free hash function to produce the hash value.

21. The method of claim 2, wherein the processor number identifies a microprocessor of the second computer system.

22. The method of claim 21, wherein the processor number uniquely identifies the microprocessor.

23. The computer system of claim 7, wherein the processor number identifies a microprocessor of the apparatus.

24. The computer system of claim 23, wherein the processor number uniquely identifies the microprocessor.

25. The article of claim 14, wherein the processor number identifies a microprocessor of the first system.

26. The article of claim 25, wherein the processor number uniquely identifies the microprocessor.